2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

# SVM Classification for Imbalanced Data Using Conformal Kernel Transformation

Yong Zhang, Panpan Fu, Wenzhe Liu, Li Zou
School of Computer and Information Technology
Liaoning Normal University
Dalian, China

***Abstract*—The problem of classifying imbalanced datasets has drawn a significant amount of interest from academia and industry. In this paper, we propose a modified support vector machine (SVM) approach using conformal kernel transformation to address the class imbalance problem. The proposed method first uses standard SVM algorithm to obtain an approximate hyperplane. And then, we give a kernel function and compute its parameters using the chi-square test. Finally, an experimental analysis is carried out with a wide range of highly imbalanced datasets over the proposal and several other methods. The results show that our proposal outperforms previously proposed methods.**

***Keywords*—support vector machine; classification; imbalanced data; conformal kernel transformation**

## I. INTRODUCTION

The scenario of imbalanced data appears in the classification field when the size of samples that represent the different categories is very different among them [1]. Imbalanced data are quite common in many real applications, such as fraud detection, medical diagnosis, network intrusion detection, and so on. For a binary classification problem, one of the classes can be represented by only a few samples while another class is represented by a large number of samples. The former is called the minority class and the later is called the majority class.

Learning from the imbalance data is a remarkable challenge in the knowledge discovery and data mining field. When dealing with imbalanced datasets, many standard learning methods tend to emphasize the majority class and ignore the minority class [2, 3]. The fundamental reason is that these methods assume relatively balanced class distributions and equal misclassification costs within imbalanced datasets [4]. Therefore, The classification performance for the minority class becomes unsatisfactory.

In recent years, the imbalanced learning problem has drawn a significant amount of interest from academia and industry. Much work focusing on this topic has been done. Many proposed methods have improved the classifier performance to some extents.

In this paper, we propose a modified support vector machine (SVM) method using conformal kernel transformation to solve the imbalanced data classification problem. First, we use standard SVM algorithm to gain an approximate hyperplane. Then, we give a kernel function and compute its parameters using the chi-square test. Finally, experimental results on 6 datasets show that under *F-measure* and *G-mean* metrics, the proposed method could achieve more classification performance when dealing with the data skewed distribution.

The remainder of this paper is organized as follows. Section II outlines the related background of imbalanced learning and support vector machine. The proposed method is presented in Section III. Experimental results are provided in Section IV. Finally we conclude in Section V.

## II. BACKGROUND

The goal of this section is to provide the background information needed to describe our proposal. It is divided in two parts: an introduction to the problem of classification with imbalanced datasets, and a briefly review of support vector machine.

### *A. Classification with Imbalanced Datasets*

The problem of classifying imbalanced datasets has been faced in many real works. While classifiers are built by the imbalanced datasets, the minority class is usually overwhelmed by the majority class [4]. Prediction results of those classifiers are dominated by the majority class. Researchers have proposed many strategies to deal with the class imbalance problem. Typically, the methods developed for coping with class imbalance can be classified into two kinds of approaches: data-level approaches and algorithmic approaches.

The goal of data-level approaches is to obtain a more or less balanced class distribution based on the idea of resampling the data. Resampling techniques can be categorized as undersampling methods, oversampling methods and hybrid methods. In order to generate a balanced dataset from the original imbalanced one, undersampling methods [5] create a subset of the original dataset by deleting some of the samples of the negative class; and relatively oversampling methods [6] generate a superset of the original dataset by replicating some of the samples of the positive class or creating new samples from the original positive class instances. Hybrid methods [7] integrate both approaches into one, deleting some of the samples after the application of the oversampling method in order to remove the induced overfitting.

Undersampling with imbalanced datasets can be considered as a prototype selection procedure with the purpose of balancing datasets to achieve a high classification rate, avoiding the bias toward majority class samples. García and

Herrera [8] propose an evolutionary undersampling method for classification with imbalanced datasets. Galar *et al*. [9] develop a new ensemble construction algorithm (EUSBoost) based on RUSBoost and evolutionary undersampling. Synthetic Minority Over-sampling Technique (SMOTE) [6] is an intelligent oversampling method using synthetic samples. SMOTE method adds new synthetic samples to the minority class by randomly interpolating pairs of the closest neighbors in the minority class. Borderline-SMOTE [10] is another approach based on the synthetic generation of instances proposed in SMOTE. In this case, only the positive samples that lie near the decision boundaries (the borderline) are used to oversample the positive class. Gao *et al*. [11] propose probability density function estimation based oversampling approach for two-class imbalanced classification problems. Cateni *et al*. [12] propose a hybrid resampling method, called SUNDO. SUNDO combines the two approaches: for the oversampling phase, it places new samples where they likely could be and avoids to place them close to frequent samples; moreover it employs an innovative undersampling technique.

Algorithmic approaches try to modify the classifiers to suit the imbalanced datasets. Cost-sensitive approach [13] is a popular algorithmic approach, which assigns a different misclassification cost (weight) for each training sample, and then minimizes the total misclassification cost. In this framework the costs of misclassifying a rare pattern are higher with respect to other kinds of errors in order to encourage their correct detection. In general, cost-sensitive approaches give penalties to misclassification in different classes, but in practical it is difficult to predefine the proper penalty for each class. Sun *et al*. [14] investigate cost-sensitive boosting algorithms for advancing the classification of imbalanced data, and propose three cost-sensitive boosting algorithms by introducing cost items into the learning framework of AdaBoost. Cano *et al*. [18] uses a matrix of weights to describe the importance of each attribute in the classification of each class, and improves the classification performance by considering both global and local data information. Zong *et al.* [15] propose a weighted extreme learning machine (ELM) to deal with imbalanced learning problem, in which each training sample is assigned with an extra weight to strengthen the impact of minority class while weaken the relative impact of majority class. Li *et al*. [16] present a Boosting weighted ELM, which embeds weighted ELM into a modified AdaBoost framework, to solve the above problem.

Many works make some modification of the classification algorithms. SVMs have also been employed for facing imbalanced datasets. Many researchers combine the weighting method with the SVM, and have proposed a variety of weighted approaches for the class imbalance learning [17, 18]. Batuwita and Palade [19] propose a method to improve fuzzy SVMs for class imbalance learning, which can be used to handle the class imbalance problem in the presence of outliers and noise. Wu and Chang [20, 21] propose a class-boundary-alignment algorithm to adjust the boundary with an alignment in kernel, and improve SVM performance in imbalanced datasets. A particular kind of radial basis function has been developed and tested in [22] where hyper-rectangular activation function neurons are used in the hidden layer in order to achieve more precision in the detection of the boundary of the input space regions reserved to each class. López *et al.* [23] propose the usage of the Iterative Instance Adjustment for Imbalanced Domains (IPADE-ID) algorithm to address imbalanced classification. The active learning approach provides more balanced training samples because it selects samples that lie closest to the separating hyperplane [24]. Fu and Lee [25] present a certainty-based active learning (CBAL) algorithm to solve the imbalanced data classification problem.

In recent years, ensemble of classifiers have arisen as a possible solution to the class imbalance problem attracting great interest among researcher because of their flexible characteristics [26]. Ensembles are designed to increase the accuracy of a single classifier by training several different classifiers and combining their decisions to output a single class label [27]. Using an ensemble of weak classifiers to boost the classification performance has been reported to be effective in skewed data. Bagging and Boosting are two of the most popular ensemble learning algorithms among them [27]. SMOTEBoost [28] is designed to alter the imbalanced distribution based on Boosting. Data generation techniques are involved to emphasize the minority class examples at each iteration of Boosting. Easyensemble method [29] is developed based on the Bagging classification, which samples several subsets from the majority class, trains a learner using each of them, and combines the outputs of those learners. Oh *et al*. [30] present an ensemble learning method combined with active example selection to resolve the imbalanced data problem. Liu *et al.* [31] propose to combine an integrated sampling technique, which incorporates both over-sampling and under-sampling, with an ensemble of SVMs to improve the prediction performance.

### B. Support Vector Machine

A classification technique that has received considerable attention is support vector machine (SVM) proposed by Vapnik [32]. SVM has its roots in statistical learning theory and has shown promising empirical results in many practical applications.

Consider a binary classification problem consisting of $N$ training samples. Each sample is denoted by a tuple ($\mathbf{x}_i$, $y_i$) ($i$=1, 2, …, $N$), where $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{id})^T$ represents an $d$-dimensional data sample, and $y_i \in \{+1, -1\}$ denotes its class label. The decision boundary of a linear classifier can be written in the following form

$$\mathbf{w} \cdot \mathbf{x} + b = 0\,, \tag{1}$$

where $\mathbf{w}$ and $b$ are parameters of the model.

The support vector technique requires the solution of the following optimization problem:

$$\min_{\mathbf{w},b,\xi} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \tag{2}$$

subject to the constraints

$$y_i(\mathbf{w}\cdot\mathbf{x}_i+b)+\xi_i-1\geq 0$$
$$\xi_i\geq 0,\qquad i=1,\cdots,N \tag{3}$$

where parameter $C$ is a user-specified positive parameter that controls the trade-off between maximizing the margin and minimizing the training error term. The slack variables $\xi_i>0$ hold for misclassified samples, and therefore, $\sum_{i=1}^{N}\xi_i$ can be thought of as a measure of the amount of misclassifications.

This quadratic-optimization problem can be solved by constructing a Lagrangian representation. Once the optimal pair $(\mathbf{w}_0,b_0)$ is determined, the SVM decision function is then given by

$$f(\mathbf{x})=sign(\sum_{i\in SV}\lambda_i y_i\mathbf{x}_i\cdot\mathbf{x}+b) \tag{4}$$

where the sample $\mathbf{x}_i$ with the corresponding non-zero $\lambda_i$ is called a support vector (*SV*). If $f(\mathbf{x})=1$, then the test sample $\mathbf{x}$ is classified as a positive class; otherwise, it is classified as a negative class.

SVM works very well with high-dimensional data and avoids the curse of dimensionality problem. Its essential idea is to use a kernel function to map the original input data into a high-dimensional space so that two classes of data become, as far as possible, linearly separable [32]. For a nonlinear SVM, the decision function is given by

$$f(\mathbf{x})=sign(\sum_{i\in SV}\lambda_i y_i K(\mathbf{x}_i,\mathbf{x})+b) \tag{5}$$

where $K(\mathbf{x}_i,\mathbf{x}_j)=\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$ is called the kernel function.

Thus, the kernel is the key that determines the performance of the SVM. Several typical kernel functions are the lineal kernel $K(\mathbf{x}_i,\mathbf{x}_j)=\mathbf{x}_i\cdot\mathbf{x}_j$, the polynomial kernel $K(\mathbf{x}_i,\mathbf{x}_j)=(a\mathbf{x}_i{}^T\mathbf{x}_j+r)^d$ and the RBF kernel $K(\mathbf{x}_i,\mathbf{x}_j)=\exp(-\gamma\|\mathbf{x}_i-\mathbf{x}_j\|^2)$.

## III. THE PROPOSED ALGORITHM

In this section, we first give a description of conformal kernel transformations, and then we present our proposed method.

### *A. Conformal Kernel Transformations*

A conformal transformation is a transformation that preserves local angles. To improve SVM discrimination power, Amari and Wu [33, 34] modified a kernel with a conformal mapping, and were the first to propose the use of conformal transformations to obtain a data-dependent kernel function. A conformal transformation of a geometrical space could be defined as a function that maps that space into a new one in which the angles between curves are locally preserved. If $c(x)$ is a positive real valued function of $x$, then a new kernel is created by

$$\tilde{k}(x,x')=c(x)k(x,x')c(x') \tag{6}$$

$$c(x)=\sum_{x_s\in SVs}e^{-m\|x-x_s\|^2} \tag{7}$$

where $m$ is a positive constant, *SV*s are support vectors, $c(x)$ is a suitably chosen positive function.

We can obtain a new kernel function $\tilde{k}(x,x')$ by a conformal transformation of the original $k(x,x')$. However, $c(x)$ in equation (7) is sensitive to the distribution of *SV*s. The parameter $m$ in equation (7) is computed through the distances, in input space, between support vectors, so it is dynamic but not well adaptive to spatial distribution in feature space. A modified version is presented in [34] which consider different $m_i$ for different *SV*s. Wu and Chang [35] also introduce a conformal transformation on kernel function, but the width is calculated through the distances in feature space.

Williams *et al.* [36] propose a kernel-scaling method and describe a more direct way of achieving the desired magnification. Its idea is to choose a function $c(x)$ so that it decays directly with distance, and to use prior knowledge obtained from conventional SVM training to conformally rescale the initial kernel function, so that the separation between two classes of data is effectively enlarged. The proposed transformation function is

$$c(x)=e^{-mf(x)^2}, \tag{8}$$

where $f(x)$ is given by equation (5) and $m$ is a positive constant. $c(x)$ reaches its maximum on the boundary surface, where $f(x)=0$, and decays smoothly to $e^{-m}$ at the margins of the separating region where $f(x)=\pm 1$.

The method proposed in [36] proved to be robust and efficient, but does not account for imbalanced data. Maratea *et al.* [37] propose an asymmetric kernel scaling (AKS) method for extending to imbalanced binary classification problem. Its basic idea is to enlarge differently areas on the two sides of the boundary surface, so to compensate for its skewness towards minority samples. The applied kernel transformation function in [37] is

$$c(x)=\begin{cases}e^{-m_1 f(x)^2}, & if\ x\in V^+\\ e^{-m_2 f(x)^2}, & if\ x\in V^-\end{cases} \tag{9}$$

where $m_1$ and $m_2$ are free parameters, $V^+$ and $V^-$ are the positives and the negatives according to the initial standard SVM prediction, respectively.

This paper uses chi-square test to compute parameter $m_i$ in order to avoid to optimizing $m_i$ and decrease computation cost, and proposes a modified SVM classification algorithm using conformal kernel transformation. First, we also give a selection of kernel function $c(x)$ using equation (9). Then, we calculate parameter $m_i$ by using the chi-square test in section *B*. Finally, we present an algorithm description in section *C*.

### *B. Computation of Parameters $m_i$*

Similar to AKS method in [37], we first perform a standard SVM to compute an approximate boundary position, then we split the samples in two datasets denoted by $V^+$ and $V^-$, according to first step prediction.

Due to inevitable relation between $m_i$ and samples of each category, this paper uses the chi-square test and weighting to calculate $m_i$.

The chi-square test is used to determine whether there is a significant difference between the expected frequencies and the observed frequencies in one or more categories. The formula for calculating chi-square ( $\chi^2$ ) is:

$$\chi^2 = \sum \frac{(f_o - f_e)^2}{f_e} \tag{10}$$

where $f_o$ is the observed frequency in each category, $f_e$ is the expected frequency in the corresponding category. That is, chi-square is the sum of the squared difference between observed data $f_o$ and the expected data $f_e$, divided by the expected data in all possible categories.

Let $V$ be a dataset including $N$ samples and binary classes. $n_i$ denotes the number of samples in the $i$th category ($i$=1, 2). In the optimal distribution, the chi-square value is

$$\chi^2 = \sum_{i=1}^{2} \frac{(n_i - N/2)^2}{N/2} \tag{11}$$

$m_i$ is defined as follows:

$$m_i = w_i \times \frac{X_i}{\chi^2} = w_i \times \frac{X_i}{\sum_{i=1}^{2} X_i}, \quad i=1, 2. \tag{12}$$

where $X_i = \frac{(n_i - N/2)^2}{N/2}$ ($i$=1, 2), $w_i$ is the weights calculated by equation (12) as follows

$$w_i = \frac{N/n_i}{\sum_{i=1}^{2} N/n_i} \quad (i=1, 2). \tag{13}$$

Setting the appropriate weight is a critical issue in weighted approach for imbalanced problem. Obviously, weight factors $w_i$ above satisfy $\sum_{i=1}^{2} w_i = 1$. $w_i$ can show sparse distribution nature of each category.

### *C. Algorithm Description*

The proposed algorithm is described as follows. In each iteration, the proposed algorithm first calculates $m_i$ for each support vector based on a chi-squared test. Then, the kernel transformation function $c(x)$ is computed. Finally, the new kernel matrix $K$ is updated and the classification model is retrained.

**Algorithm:** A modified SVM algorithm
**Input:**
The training set $X_{train}$ ; the kernel matrix $K$; maximum running iterations $T$.
**Output:**
Classifier $svm_T$.
**Begin**
1: Train a SVM $svm_0$ with kernel matrix $K_0 = K$ and the training set $X_{train}$ .
2: Compute the distance $f(x)$ from sample $x \in X_{train}$ to the approximate hyperplane, and obtain an initial data partition $V^+$ and $V^-$, where represent the positive dataset and the negative dataset, respectively.
3: $t \leftarrow 0$.
4: **while** ($t<T$) {
5: Compute $m_i$ using equation (12).
6: Compute kernel transformation function $c(x)$ using equation (9).
7: Compute the new kernel matrix $K$ using equation (6).
8: Train a new SVM $svm_t$ with kernel matrix $K$ and amend the approximate hyperplane.
9: $t \leftarrow t+1$. }
10: **return** $svm_T$.
**End**

## IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we give a description of evaluation metrics for imbalanced data classification problem, and present the experimental analysis of the proposed method in order to determine its robustness in imbalanced datasets.

### *A. Evaluation Metrics*

Overall accuracy and error rate are important evaluation metrics for assessing the classification performance and guiding the classifier modeling. However, in the case of imbalanced learning, conventional evaluation metrics fail to provide adequate information about the performance of the classifier [38].

For example, if we have a binary classification problem, where 99% of samples belong to the majority class, and the rest belong to the minority class. A classifier assigns all samples to the majority class to easily achieve 99% accuracy. However, this measurement is meaningless to some applications where the learning concern is the identification of the rare cases.

The measures of the quality of classification are built from a confusion matrix shown in Table I. The result of classification can be categorized into four cases. These categories are called *TP* (true positive), *FN* (false negative), *FP* (false positive) and *TN* (true negative).

TABLE I. CONFUSION MATRIX FOR A BINARY-CLASS PROBLEM

| Actual class | Positive prediction | Negative prediction |
|---|---|---|
| Positive class | TP (True Positive) | FN (False Negative) |
| Negative class | FP (False Positive) | TN (True Negative) |

From this matrix, different measures can be deduced to perform the evaluation in the imbalanced data classification problem:

- *True positive rate* or *sensitivity*, $TP_{rate}=Sens=TP/(TP+FN)$.
- *Ture negative rate* or *specificity*, $TN_{rate}=Spec=TN/(FP+TN)$.
- *False positive rate*, $FP_{rate}=FP/(FP+TN)$.
- *False negative rate*, $FN_{rate}=FN/(TP+FN)$.

Accuracy is the most used evaluation metric for assessing the classification performance and guiding the classifier modeling. The overall accuracy *Acc* is defined as

$$Acc=\frac{TP+TN}{TP+FP+TN+FN} \tag{14}$$

However, accuracy is not a useful measure for imbalanced data. To measure the performance of the classifier, several measures have been developed to deal with the class imbalance problem, including *F-measure* and geometric mean (*G-mean*) [39].

Precision *Prec* is defined as the fraction of relevant instances that are retrieved as follows:

$$Prec=TP/(TP+FP). \tag{15}$$

*F-measure* is often used in the fields of information retrieval and machine learning for measuring search, document classification, and query classification performance. *F-measure* considers both precision *Prec* and sensitivity *Sens* to compute the score [40]. It can be interpreted as a weighted average of the precision and sensitivity as follows:

$$F-measure=\frac{2\times Prec\times Sens}{Prec+Sens}. \tag{16}$$

*G-mean* is defined by two parameters sensitivity *Sens* and specificity *Spec*. *G-mean* gives a more fair comparison of positive class and negative class regardless of its size. It is defined as

$$G-mean=\sqrt{Sens\times Spec}. \tag{17}$$

### B. Experimental comparison and anlysis

In this sub-section, we will compare our proposal to address the class imbalance problem with several techniques. The experiments use 6 binary-class datasets which have different degrees of imbalance from the KEEL [41], including *pima*, *haberman*, *glass*1, *cmc*1, *yeast*1, and *yeast*2. They are very varied in their size of classes, size of attributes, size of samples and imbalance ratio. Their characteristics are summarized in Table II. For each data set, the size of samples (#Samples), the size of attributes (#Attributes), the size of samples of each class (#Positives and #Negatives), and imbalance ratio are listed. We calculated class imbalance ratio of the size of the majority class to the size of the minority class. Table II is ordered by the imbalance ratio, from lowly to highly imbalanced datasets.

TABLE II. DATA DESCRIPTION FOR BINARY PROBLEM

| Data set | #Samples | #Attributes | #Positives and #Negatives | Imbalance ratio |
|---|---|---|---|---|
| pima | 691 | 8 | 241, 450 | 1.87 |
| haberman | 306 | 3 | 81, 225 | 2.78 |
| glass1 | 214 | 9 | 51, 163 | 3.20 |
| cmc1 | 1324 | 9 | 299, 1025 | 3.43 |
| yeast1 | 1332 | 8 | 146, 1186 | 8.12 |
| yeast2 | 1332 | 8 | 84, 1248 | 14.86 |

In these datasets, *pima* is a binary-class dataset. For *haberman* dataset, we select the first class as positive class and the second class as negative class. For *glass*1 dataset, the 4th, 5th and 6th classes in original data are integrated into positive class and the rest is negative class. For *cmc*1 dataset, we select the second class in original data as positive classes and the rest is negative class. For *yeast*1 data set, we use the 4th class in original data as positive class and the rest is negative class. For *yeast*2 data set, we use the 5th and 6th classes in original data as positive class and the rest is negative class.

In order to evaluate the performance of our proposed solution, we have compared it against other popular classifiers, including AKS method [37], SVM-SMOTE, SVM-UNDER, and WSVM in binary classification problems. SVM-SMOTE is a SMOTE-based SVM classifier. SVM-UNDER is a undersampling-based SVM classifier. WSVM is a weighting-based SVM classifier.

In the experiments, we implemented all 5 algorithms in Matlab. Parameters of each algorithm are tuned using gird search with 10-cross validation. For each dataset, we randomly select 80% of data as training data while the rest are used as test data.

#### 1) F-measure and G-mean metrics

*F-measure* and *G-mean* values of two models are calculated from the accuracies of each class according to equations (16) and (17). The experimental results are presented in Table III.

From Table III, we observe that, for most datasets, our proposed method and AKS method perform better than SVM-SMOTE, SVM-UNDER, and WSVM, with respect to both *F-measure* and *G-mean*. However, WSVM method performs the best *G-mean* among all methods. For most datasets, our proposed method performs slightly better that AKS method. Indeed, the performance improvements of our proposed method are significant when comparing to AKS method on datasets *haberman* and *glass*1.

TABLE III. AVERAGE PERFORMANCE IN 6 DATASETS

| Data set | SVM-SMOTE | | SVM-UNDER | | WSVM | | AKS method | | Our proposed method | |
|---|---|---|---|---|---|---|---|---|---|---|
| | F-measure | G-mean | F-measure | G-mean | F-measure | G-mean | F-measure | G-mean | F-measure | G-mean |
| pima | 0.342 | 0.472 | 0.690 | 0.760 | 0.733 | 0.797 | 0.752 | 0.813 | 0.824 | 0.843 |
| haberman | 0.333 | 0.461 | 0.348 | 0.501 | 0.583 | 0.695 | 0.586 | 0.568 | 0.830 | 0.822 |
| glass1 | 0.845 | 0.912 | 0.780 | 0.839 | 0.923 | 0.970 | 0.801 | 0.817 | 0.952 | 0.961 |
| cmc1 | 0.290 | 0.486 | 0.381 | 0.572 | 0.270 | 0.432 | 0.572 | 0.538 | 0.607 | 0.612 |
| yeast1 | 0.800 | 0.894 | 0.781 | 0.901 | 0.857 | 0.928 | 0.912 | 0.934 | 0.926 | 0.981 |
| yeast2 | 0.462 | 0.615 | 0.474 | 0.745 | 0.518 | 0.713 | 0.713 | 0.754 | 0.847 | 0.885 |

### 2) *Performance result in terms of accuracy*

Performance result in terms of accuracy is shown in Table IV. We see that our proposed method outperforms the others for most datasets. This is consistent with our analysis in *F-measure* and *G-mean* metrics. In other words, our proposed method performs best in achieving higher overall accuracy, while AKS method obtains relatively slight decrease in the overall accuracy.

TABLE IV. OVERALL ACCURACY IN 6 DATASETS

| Data set | SVM-SMOTE | SVM-UNDER | WSVM | AKS method | Our proposed method |
|---|---|---|---|---|---|
| pima | 0.649 | 0.766 | 0.792 | 0.799 | 0.831 |
| haberman | 0.750 | 0.531 | 0.781 | 0.823 | 0.865 |
| glass1 | 0.913 | 0.891 | 0.956 | 0.926 | 0.953 |
| cmc1 | 0.638 | 0.563 | 0.745 | 0.781 | 0.791 |
| yeast1 | 0.954 | 0.941 | 0.961 | 0.965 | 0.967 |
| yeast2 | 0.909 | 0.869 | 0.931 | 0.949 | 0.954 |

## V. CONCLUSIONS

In this paper, a modified SVM method is proposed to deal with the imbalanced data classification problem. In the proposed method, a kernel transformation function is applied, and its parameters are calculated by the chi-square test and weighting. Experimental results show our proposal outperforms previously proposed methods.


## ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (Nos. 61373127, 61105059), the China Postdoctoral Science Foundation (No. 20110491530), and the University Scientific Research Project of Liaoning Education Department of China (No. 2011186).


## REFERENCES


[1] N. V. Chawla, "Data mining for imbalanced datasets: an overview," in: Data Mining and Knowledge Discovery Handbook, Springer, 2010, pp. 875–886.

[2] S. Daskalaki, I. Kopanas, and N. Avouris, "Evaluation of classifiers for an uneven class distribution problem," Applied Artificial Intelligent, vol. 20, no. 5, pp. 381–417, 2006.

[3] N. V. Chawla, N. Japkowicz, and A. Kolcz, "Editorial: special issue on learning from imbalanced data sets," SIGKDD Explorations, vol. 6, no. 1, pp. 1–6, 2004.

[4] H. B. He and E. A. Garcia, "Learning from imbalanced data," IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263–1284, 2009.

[5] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: one-sided selection," in: Proceedings of the International Conference on Machine Learning, 1997, pp.179–186.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, vol. 16, no. 1, pp. 231–357, 2002.

[7] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behaviour of several methods for balancing machine learning training data," SIGKDD Explorations, vol. 6, no.1, pp. 20–29, 2004.

[8] S. García and F. Herrera, "Evolutionary undersampling for classification with imbalanced datasets: proposals and taxonomy," Evolutionary Computation, vol. 17, pp. 275–306, 2009.

[9] M. Galar, A. Fernández, E. Barrenechea, and F. Herrera, "EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling," Pattern Recognition, vol. 46, no. 12, pp. 3460–3471, 2013.

[10] H. Han, W. Wang, and B. Mao, "Borderline-SMOTE: a new over-sampling method in imbalanced datasets learning," in: Proceedings of the 2005 International Conference on Intelligent Computing (ICIC'05), Lecture Notes in Computer Science, vol. 3644, 2005, pp.878–887.

[11] M. Gao, X. Hong, S. Chen, and C. J. Harris, "Probability density function estimation based over-sampling for imbalanced two-class problems," in: Proceedings of the 2012 International Joint Conference on Neural Networks (IJCNN 2012), 2012.

[12] S. Cateni, V. Colla, and M. Vannucci, "A method for resampling imbalanced datasets in binary classification tasks for real-world problems," Neurocomputing, vol. 135, pp. 32–41, 2014.

[13] B. X. Wang and N. Japkowicz, "Boosting support vector machines for imbalanced data sets," Knowledge and Information Systems, vol. 25, no. 1, pp. 1–20, 2000.

[14] Y. Sun, M. S. Kamel, A. K. C. Wong, and Y. Wang,"Cost-sensitive boosting for classification of imbalanced data," Pattern Recognition, vol. 40, no. 12, pp. 3358–3378, 2007.

[15] W. Zong, G. B. Huang, and Y. Chen, "Weighted extreme learning machine for imbalance learning," Neurocomputing, vol. 101, pp. 229–242, 2013.

[16] K. Li, X. F. Kong, Z. Lu, W. Y. Liu, and J. P. Yin, "Boosting weighted ELM for imbalanced learning," Neurocomputing, vol. 128, pp. 15–21, 2014.

[17] X. Yang, Q. Song, and Y. Wang, "A weighted support vector machine for data classification," International Journal of Pattern Recognition and Artificial Intelligence, vol. 21, no. 5, pp. 961–976, 2007.

[18] A. Cano, A. Zafra, and S. Ventura, "Weighted data gravitation classification for standard and imbalanced data, " IEEE Transactions on Cybernetics, vol. 43, no. 6, pp. 1672-1687, 2013.

[19] R. Batuwita and V. Palade, “FSVM-CIL: fuzzy support vector machines for class imbalance learning,” IEEE Transactions on Fuzzy Systems, vol. 18, no. 3, pp. 558–571, 2010.

[20] G. Wu and E. Y. Chang, “Class-boundary alignment for imbalanced data set learning,” In: ICML 2003 workshop on learning from imbalanced data sets, 2003, pp. 49–56.

[21] G. Wu and E. Y. Chang, “KBA: kernel boundary alignment considering imbalanced data distribution,” IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, pp. 786–795, 2005.

[22] V. Soler and M. Prim, “Rectangualr basis functions applied to imblanced datasets,” Lecture Notes in Computer Science, vol. 4668, 2007, pp. 511–519.

[23] V. López, I. Triguero, C. J. Carmona, S. García, and F. Herrera, “Addressing imbalanced classification with instance generation techniques: IPADE-ID,” Neurocomputing, vol. 126, pp. 15–28, 2014.

[24] S. Ertekin, J. Huang, L. Bottou, and C. L. Giles, “Learning on the border: active learning in imbalanced data classification,” in: Proc. Of ACM Conf. Information and Knowledge Management (CIKM ’07), pp. 127–136, 2007.

[25] J. H. Fu and S. L. Lee, “Certainty-based active learning for sampling imbalanced data sets,” Neurocomputing, vol. 119, pp. 350–358, 2013.

[26] G. B. Huang, Q. Y. Zhu, and C. K. Siew, “Extreme learning machine: Theory and applications,” Neurocomputing, vol. 70, nos. 1-3, pp. 489–501, 2006.

[27] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, “A review on ensembles for the class imbalance problem bagging-, boosting-, and hybrid-based approaches,” IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 42, no. 2, pp. 463–484, 2012.

[28] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowye, “SMOTEBoost: improving predic-tion of the minority class in boosting,” in: Proceedings of Knowledge discovery in databases (PKDD 2003), pp. 107–19, 2003.

[29] X. Y. Liu, J. X. Wu, and Z. H. Zhou, “Exploratory undersampling for class-imbalance learning,” IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, vol. 39, no. 2, pp. 539–550, 2009.

[30] S. Oh, M. S. Lee, and B. T. Zhang, “Ensemble Learning with active example selection for imbalanced biomedical data classification,” IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 8, no. 2, pp. 316–325, 2011.

[31] Y. Liu, X. H. Yu, J. X. Huang, and A. J. An, “Combining integrated sampling with SVM ensembles for learning from imbalanced data sets,” Information Processing and Management, vol. 47, no. 4, pp. 617–631, 2011.

[32] V. Vapnik, “The Nature of Statistical Learning Theory,” New York: Springer-Verlag, 1995.

[33] S. Amari and S. Wu, “Improving SVM classifiers by modifying kernel functions,” Neural Networks, vol. 12, no. 6, pp. 783–789, 1999.

[34] S. Wu and S. Amari, “Conformal transformation of kernel functions: a data-dependent way to improve support vector machine classifiers,” Neural Processing Letters, vol. 15, pp. 59–67, 2001.

[35] G. Wu and E. Y. Chang, “Adaptive feature-space conformal transformation for imbalanced-data learning,” in: Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington, DC, 2003, pp. 816–823.

[36] P. Williams, S. Li, J. Feng, and S. Wu, “Scaling the kernel function to improve performance of the support vector machine,” Lecture Notes in Computer Science, vol. 3496, pp. 251–312, 2005.

[37] A. Maratea, A. Petrosino, and M. Manzo, “Adjusted F-measure and kernel scaling for imbalanced data learning,” Information Sciences, vol. 257, pp. 331–341, 2014.

[38] M. V. Joshi, V. Kumar, and R. C. Agarwal, “Evaluating boosting algorithms to classify rare classes: comparison and improvements,” in: Proceedings of the 1st IEEE International Conference on Data Mining (ICDM’ 01), pp. 257–264, December 2001.

[39] T. Fawcett, “ROC graphs: Notes and practical considerations for researchers,” HP Labs, Palo Alto, CA, Tech. Rep. HPL-2003-4, 2003.

[40] D. Lewis and W. Gale, “Training text classifiers by uncertainty sampling,” in: Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information, New York, NY, 1998, pp. 73–79.

[41] J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, “KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework,” Journal of Multiple-Valued Logic and Soft Computing, vol. 17, nos. 2-3, pp. 255–287, 2011.